Chair for Algebraic and Logical
Foundations of Computer Science
TU Dresden

Winter Term 2021/2022

Logic Programming Engineering – Assignment 1

Dr.-Ing. Sascha Klüppelholz

Exercise 1.1 [Simple arithmetics, relations, and functions in Prolog]

What will be the result of the following queries? Please fill the table by first writing down your predicted answer and then the actual answer when typing the commands into SWI-Prolog. Compare the results and find explanations whenever the outcome differs from what your prediction was.

query	Prologs' answer
`4 + 3 = 7.`	`False.`
`4 + 3 =:= 7.`	`True.`
`X is 4 + 3, X = 7.`	`X = 7.`
`7 is 4 + 3.`	`True.`
`4 + 3 is 7.`	`False.`
`1 is 0.5 + 0.5.`	`False.`
`1.0 is 0.5 + 0.5.`	`True.`
`X is 2*(3-1+18)/2**4.`	`X = 2.5.`
`X is 2*(3-1+18)//2**4.`	`X = 2.`
`X is div(2*(3-1+18),2**4).`	`X = 2.`
`X is mod(2*(3-1+18),2**4).`	`X = 8.`
`X is rdiv(12,18).`	`X = 2 rdiv 3.`
`X is 4 rdiv 3 + 1.`	`X = 7 rdiv 3.`
`X is rdiv(1,3) + rdiv(7,8).`	`X = 29 rdiv 24.`
`X is 4 rdiv 3 + 1.5.`	`X = 2.833333333333333.`
`X is 4 rdiv 3 + rational(1.5).`	`X = 17 rdiv 6.`
`between(2,4,3).`	`True.`
`between(2,4,B).`	`B = 2 ; B = 3 ; B = 4.`
`between(2,inf,B).`	`B = 2 ; B = 3 ; B = 4 ; B = 5 ; ...`
`plus(1,2,P), succ(P,X).`	`P = 3, X = 4.`
`X is random(7).`	`X = 2.`
`random(X).`	`X = 0.6506324665549162.`

Exercise 1.2 [Distance of two points in the 2-dimensional plane]

Provide a Prolog predicate `distance/3` to calculate the distance between two points in the 2-dimensional plane. The points are given as pairs of coordinates.

Solution:

```
distance((X1,Y1),(X2,Y2),Z):-
    number(X1), number(Y1), number(X2), number(Y2),
    Z is (sqrt((abs(X2-X1)**2) + (abs(Y2-Y1)**2))).
```

Exercise 1.3 [Euclid's algorithm]

a) Implement the naive version of Euclid's algorithm for computing the greatest common divisor (GCD) of two non-negative integers, according to the following recursive definition:

$$\gcd(a,b) \stackrel{\text{def}}{=} \begin{cases} a & \text{if } a = b \\ \gcd(a-b,b) & \text{if } a > b \\ \gcd(a,b-a) & \text{if } a < b \end{cases}$$

The new predicate should be called `gcd/3` and, given two non-negative integers in the first two argument positions, the third position should match with the GCD of the two given numbers.

Solution:

```
gcd(A,A,A) :- !.

gcd(A,B,E) :-
  A > B, !,
  A1 is A - B,
  gcd(A1,B,E).

gcd(A,B,E) :-
  A < B,
  B1 is B - A,
  gcd(A,B1,E).
```

b) Provide a predicate `euclid/3` with the same interface and usage as `gcd/3` that uses modulo rather than repeated subtraction, according to the following alternative recursive definition of the GCD:

$$\gcd(a,b) \stackrel{\text{def}}{=} \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \bmod b) & \text{else} \end{cases}$$

Solution:

```
euclid(A,0,A) :- !.

euclid(A,B,E) :-
    X is mod(A,B),
    euclid(B,X,E).
```

c) Provide a predicate `ext_euclid/3` for the extended Euclidian algorithm that allows computing the triple $\langle g, s, t\rangle$ with $g = \gcd(a, b) = s \cdot a + t \cdot b$. The new predicate can be defined according to the following recursive definition:

$$\text{ext_gcd}(a, b) \stackrel{\text{def}}{=} \begin{cases} \langle a, 1, 0\rangle & \text{if } b = 0 \\ \left\langle g', t', s' - t' \cdot \left\lfloor \frac{a}{b} \right\rfloor \right\rangle \text{ with } \langle g', s', t'\rangle = \text{ext_gcd}(b, a \bmod b) & \text{else} \end{cases}$$

Solution:

```
ext_euclid(A,0,(G,S,T)) :-
    (G,S,T) = (A,1,0), !.

ext_euclid(A,B,(G,S,T)) :-
    X is mod(A,B),
    ext_euclid(B,X,(GPrime,SPrime,TPrime)),
    G is GPrime,
    S is TPrime,
    T is SPrime - (TPrime*(A//B)).
```

d) Provide a Prolog predicate `lcm/3` for the least common multiple (LCM), where $\text{lcm}(a, b) \stackrel{\text{def}}{=} \frac{a \cdot b}{\gcd(a,b)}$.

Solution:

```
lcm(A,B,R) :-
    gcd(A,B,X),
    R is (B*A/X).
```